



**DMP ELECTRONICS INC.**  
**瞻營全電子股份有限公司**

No.15, Wugong 5th Rd. Xinzhuang Dist., New Taipei City, #24890, Taiwan, R.O.C. TEL : 886-2-2298-0770 FAX : 886-2-2299-1883 <http://www.dmp.tw>

# Buildroot for Vortex86EX

(2016/04/20)

# Table of Contents

Introduction.....	1
Prepare.....	1
Install Virtual Machine (optional).....	1
Install Buildroot.....	3
Config Buildroot for VEX-SOM (optional).....	4
Generate Linux kernel and rootfs image.....	4
Use Buildroot toolchain for Vortex86EX.....	5
Create the USB Flash Drive (USBLINUX).....	6
Use the USBLINUX on Vortex86EX.....	7
Access the files in PC from USBLINUX.....	7
Install Linux kernel, rootfs image, and bootloader.....	8
Finish.....	8

# Introduction

Buildroot is a software tool to generate toolchain, Linux Kernel, and rootfs for Embedded Linux system. User can use Buildroot to customize the tools and features for his own Embedded Linux system by it's simple UI. ( For the detail about Buildroot, please refer to Buildroot user manual on it's official website)

From this document, we will show you how to make a simple Embedded Linux by Buildroot, and use USB Flash Drive to install the Embedded Linux on Vortex86EX platform.

## Prepare

- One PC with internet connected.
- One Vortex86EX platform. ( This demo use ICOP VEX-6254)

```
ICOP VEX-6254 brief specification:  
DMP SoC Vortex86EX  
128MB DDR3 onboard  
Support microSD card or 512MB eMMC
```

- One USB Flash Drive.
- One microSD.  
(ICOP VEX-6254 support either microSD or eMMC depends on the different model, please use onboard eMMC if there is no microSD)
- One LAN cable  
(ICOP VEX-6254 should in the same network with PC)

## Install Virtual Machine (optional)

Buildroot only works under Linux, if you are using a Windows PC, please install the Virtual Machine and then install Linux distribution. We suggest to use Debian 7 (or later version) for development.

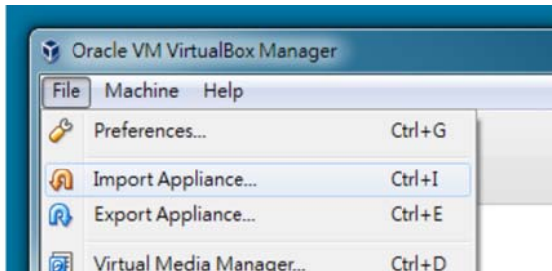
1. Download and install Virtual Machine ( we use Virtualbox as example).


VirtualBox Website: <<https://www.virtualbox.org/>>

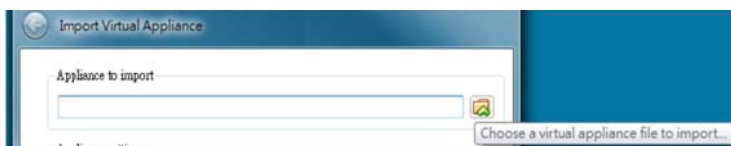
2. Download the file for VM application.

ftp://vxex:gc301@ftp.dmp.com.tw/Software/Linux/debian8.3.0\_vexsom-standard\_develop\_Env.ova

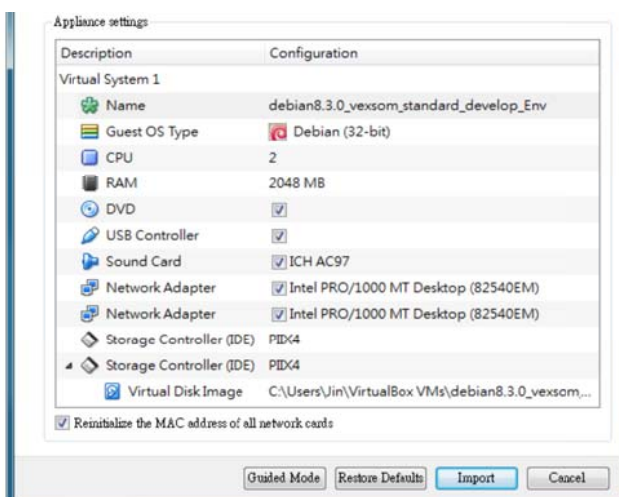
3. Open [ Oracle VM VirtualBox ], Click [ file ] -> [ Import appliance... ] .



4. Click the  icon, choose the debian8.3.0 vexsom-standard develop Env.ova, then open the file.

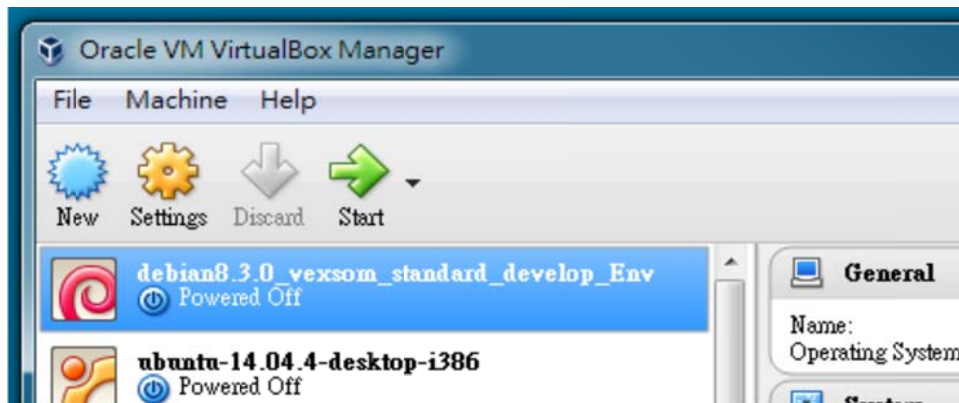


5. Modify the VM setting ( Name, CPUs, RAM....etc) according to your PC's specification or click the Import directly to use the default setting.



6. When the 'Import' is completed, you will see the new imported Virtual Machine on the left side of [ Oracle VM VirtualBox Manager ] . Please click that to open the selected VM.

After user login, you can 'make' the image directly. (Please refer to the chapter in this document called 'Generate Linux kernel and rootfs image' ).



## Install Buildroot

1. We assume working directory is in \$HOME.

```
$ cd $HOME
```

2. Download buildroot-2015.02.tar.gz.

```
$ wget ftp://vxex:gc301@ftp.dmp.com.tw/Software/Linux/buildroot-2015.02.tar.gz
```

3. Download buildroot-vortex86ex.tar.gz.

```
$ wget ftp://vxex:gc301@ftp.dmp.com.tw/Software/Linux/buildroot-vortex86ex.tar.gz
```

4. Uncompress the download files.

```
$ tar -xf buildroot-2015.02.tar.gz  
$ tar -xf buildroot-vortex86ex.tar.gz
```

5. Rename buildroot directory.

```
$ mv buildroot-2015.02 buildroot-2015.02-vexsom
```

6. Export the necessary environment variables.

```
$ export BR2_EXTERNAL=$HOME/buildroot-external  
$ export BR2_DL_DIR=$HOME/downloads
```

7. Change directory to buildroot-2015.02-vexsom, then adapt the defconfig of vexsom.

```
$ cd buildroot-2015.02-vexsom
$ make vexsom_defconfig
```

## Config Buildroot for VEX-SOM (optional)

Buildroot provides many configuration options, you can use 'make menuconfig' to start the configuration menu. The items in 'Target packages' can be selected for installing to target platform. If you don't need to config the Buildroot, please skip this chapter directly.

```
$ make menuconfig
```

Please run 'make linux-menuconfig' to modify the Linux Kernel configuration.

```
$ make linux-menuconfig
```

Please run 'make busybox-menuconfig' to modify the Busybox configuration.

```
$ make busybox-menuconfig
```

If you need to add extra files in 'rootfs' , please copy files to the 'fs\_overlay' directory. After the 'make' is completed, Buildroot will copy or overwrite the files in 'fs\_overlay' to 'rootfs' automatically .

Below is the demo for installing 'hello' to '/usr/bin/' directory in 'rootfs'.

```
$ mkdir -p $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin
$ cp hello $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin/
```

## Generate Linux kernel and rootfs image

After complete configuration, you can go for 'make', it will take time for 'make' process depends on the performance of your PC. When Buildroot finish everything, the image will be in the 'output/images/' directory.

```
$ make
$ ls output/image
bzImage rootfs.tar rootfs.tar.gz
```

In the 'output/images/' directory, you will need two files which are Linux kernel image and rootfs image:

'bzImage' : Linux kernel image

'rootfs.tar.gz' : rootfs image (contains all files for root filesystem)

## Use Buildroot toolchain for Vortex86EX

After Buildroot 'make' the image, all the files related to toolchain will be put in 'output/host/' directory, for example gcc is at 'output/host/usr/bin/i686-buildroot-linux-gnu-gcc', and toolchain sysroot will be in 'output/host/usr/i686-buildroot-linux-gnu/sysroot/' ( Buildroot has a link at output/staging/ ).

Below is the demo for how to use Buildroot toolchain to compile the hello.c program, and install to root filesystem.

1. Set the PATH

```
$ export PATH=$PATH:$HOME/buildroot-2015.02-vexsom/output/host/usr/bin
```

2. Create and compile 'hello.c'

```
$ cat >hello.c <<EOF
#include <stdio.h>
int main()
{
    printf("Hello Buildroot!\n");
    return 0;
}
EOF

$ i686-buildroot-linux-gnu-gcc -o hello hello.c
```

3. Put the hello program in 'fs\_overlay', and then 'make' again.

```
$ mkdir -p $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin
$ cp hello $HOME/buildroot-external/board/icop/vexsom/fs_overlay/usr/bin/
$ make
```

4. After 'make' finished, the hello program will be in the 'target/usr/bin/ directory' and 'rootfs.tar.gz'.

## Create the USB Flash Drive (USBLINUX)

As the images are ready, below will show you how to create a bootable USB Flash Drive with Linux kernel and rootfs images.

1. Change directory to \$HOME.

```
$ cd $HOME
```

2. Uncompress buildroot-2015.02.tar.gz.

```
$ tar -xf buildroot-2015.02.tar.gz
```

3. Rename buildroot directory.

```
$ mv buildroot-2015.02 buildroot-2015.02-usblinux
```

4. Export the necessary environment variables.

```
$ export BR2_EXTERNAL=$HOME/buildroot-external  
$ export BR2_DL_DIR=$HOME/downloads
```

5. Change directory to buildroot-2015.02-usblinux, adapt the defconfig of usblinux, then make the image.

```
$ cd buildroot-2015.02-usblinux  
$ make usblinux_defconfig  
$ make
```

6. After the 'make' is completed, you will see the 'usblinux.iso' in 'the output/images' directory.

```
$ ls output/images  
bzImage rootfs.cpio rootfs.cpio.gz syslinux usblinux.iso
```

7. Plug the USB Flash Drive to PC, mount it in Virtual Machine, then you can use 'dd' to write the usblinux.iso to USB Flash Drive. (Assume the USB Flash Drive is at '/dev/disk/by-label/usbdrive')



```
$ dd if=output/images/usblinux.iso of=/dev/disk/by-label/usbdrive bs=4k
```

8. Please take the USB Flash Drive from the PC.

## Use the USBLINUX on Vortex86EX

Please plug-in the microSD and USB Flash Drive then power-on the VEX-6254. It will automatically set the IP and Netmask (as shown below) of the first ethernet interface then start the SSH server.

```
IP      : 192.168.100.100
Netmask: 255.255.255.0
```

If the PC and VEX-6254 are on the same Network, you can login the USBLINUX from PC by SSH client.

```
$ ssh root@192.168.100.100
( password : vortex86ex )
root@usblinux:~#
```

## Access the files in PC from USBLINUX

The USBLINUX does not support NFS but support SSHFS and CurlFtpFS, you can use SSHFS or CurlFtpFS to mount the SSH server or FTP server to the directory you need.

```
(hostip is the IP for PC)
root@usblinux:~# mkdir /mnt/host
root@usblinux:~# sshfs user@hostip:~/buildroot-vexsom/output/images /mnt/host
or
root@usblinux:~# curlftpfs ftp://hostip/buildroot-vexsom/output/images
/mnt/host
```

# Install Linux kernel, rootfs image, and bootloader

Here will show how to use the automatic installation program in USBLINUX. The program will automatically format the microSD as a single partition, create ext3 filesystem in this partition, install Linux kernel, root filesystem, and then install extlinux (syslinux) bootloader.

Before the automatic installation program running, please copy Linux kernel and rootfs image in ~/ directory. In the example below, we assume 'output/images' directory on PC is mounted to the '/mnt/host' directory on USBLINUX (ICOP VEX-6254) .

```
root@usblinux:~# cp /mnt/host/bzImage ~/
root@usblinux:~# cp /mnt/host/rootfs.tar.gz ~/
```

Also, you need to assign the target disk for installation. Usually the microSD should be the '/dev/sda' ( The first Block device which the Kernel detected). Different platform might have different name, you can use 'fdisk -l' or 'lsblk -S' command to confirm it.

Finally, please run 'make-target' then the installation process will be done automatically.

```
root@usblinux:~# /opt/bin/make-target /dev/sda
```

## Finish

After complete 'make-target' installation process, your own Embedded Linux images are installed on microSD. Now you can remove the USB Flash Disk, and reset the power to enjoy your Embedded Linux.

If you have any question, please contact <[soc@dmp.com.tw](mailto:soc@dmp.com.tw)> by e-mail.